AMENDMENTS TO THE CLAIMS

1. (Currently Amended) A system for classifying packets based on packet content, the system comprising:

a sequencer operable to receive packets and to identify packet flows;

a content engine interfaced with the sequencer to receive packets and to search packet contents for predetermined expressions in a packet or in a packet flow; and

a tag map interfaced with the content engine and operable to ~~tag~~ perform mappings between expressions and subexpressions to generate modified tags for packets according to the predetermined expressions found by the content engine.

2. (Currently Amended) The system of Claim 1 wherein the sequencer comprises:

an enqueue engine operable to read packet flow sequencing information;

a packet flow tracker interfaced with the enqueue engine and operable to track packet flows with the sequencing information; and

a dequeue engine interfaced with the packet flow tracker and the content engine, the dequeue engine forwarding packets to the content engine according to ~~sequencing~~ stream identification information received from the content engine.

3. (Original) The system of Claim 2 wherein the enqueue engine is further operable to determine that a packet is out of order for that packet's flow and to transmit the out-of-order packet to have any missing packets resent.

4. (Currently Amended) The system of Claim 2 wherein the dequeue engine forwards the next packet of the flow for the ~~sequencing~~ stream identification information received from the content engine.

5. (Original) The system of Claim 4 wherein the dequeue engine determines that no packets for the packet flow are ready and determines a second packet flow to send to the content engine.

6. (Original) The system of Claim 1 wherein the content engine comprises:

a non-deterministic finite automata engine operable to search packet content for one or more regular expressions; and

one or more hash engines operable to search packet content for one or more subexpressions.

7. (Original) The system of Claim 6 further comprising a lexical analyzer interfaced with the non-deterministic finite automata engine and the hash engine, the lexical analyzer determining characters of the packets.

8. (Original) The system of Claim 6 wherein the non-deterministic finite automata engine comprises field programmable gate arrays.

9. (Original) The system of Claim 6 further comprising a state store module interfaced with the non-deterministic finite engine and operable to save the state of the non-deterministic finite automata engine associated with a packet flow so that the saved state is available for the search of the next packet of the packet flow.

10. (Original) The system of Claim 6 further comprising a tag map interfaced with the content engine to map the packet to a tag based on the content search.

11. (Currently Amended) A method for classifying packets based on content, the method comprising:

identifying packet flows;

searching packet content across the identified packet flows to find one or more predetermined regular expressions;

computing a hash for predetermined strings of the regular expressions to find one or more subexpressions; and

<u>using a tag map to:</u>

<u>perform a mapping between said regular expressions and said subexpressions; and</u>

generate a modified tag corresponding to matches between predetermined expressions and subexpressions; and

tagging packets ~~based on regular expression and subexpression matches~~ with said modified tags.

12. (Original) The method of Claim 11 wherein the packet flow comprises a TCP stream.

13. (Original) The method of Claim 12 wherein identifying packet flows further comprises:

determining if a packet is out of order;

transmitting the out of order packet to its client to have missing packets resent;

buffering the out-of-order packet until the missing packet is received; and

making the packet flow associated with the missing packet available for content searching.

14. (Original) The method of Claim 11 wherein searching further comprises finding regular expression matches by encoding the regular expressions as non-deterministic finite automata.

15. (Original) The method of Claim 14 further comprising:

computing a hash for a subexpression of a regular expression match; and

finding a subexpression match if the computed hash matches a hash in a hash look-up table.

16. (Original) The method of Claim 14 wherein the non-deterministic finite automata is encoded with field programmable gate arrays.

17. (Original) The method of Claim 16 further comprising:

storing the state of the field programmable gate arrays for a packet of a first packet flow;

searching the content of a packet of a second packet flow with the field programmable gate arrays;

loading the stored state of the first packet flow into the field programmable gate arrays; and

searching the next packet of the first packet flow.

18. (Currently Amended) A system for sequencing packet streams for content classification, the system comprising:

an enqueue engine that receives the streams and reads the stream identification of stream packets to determine if a packet is out of order;

a stream tracker interfaced with the enqueue engine that associates packets to streams based upon the stream identification read by the enqueue engine;

a dequeue engine interfaced with the stream tracker and operable to forward packets for classification based on the packets' stream identification; and

a packet buffer interfaced with the enqueue engine for storing packets, wherein the enqueue engine is operable to:

transmit an out-of-order packet so that missing packets can be resent;

mark the out-of-order packet as sent; and

buffer the out-of-order packet.

19. (Cancelled)

20. (Original) The system of Claim 18 wherein the dequeue engine is further operable to receive a stream identification, forward the next packet of the stream associated with the stream identification if the stream tracker indicates the next packet is ready, and forward the next packet of a second stream if the next packet of the associated stream is not ready.

21. (Original) The system of Claim 20 wherein the dequeue engine updates the stream tracker to indicate when a packet is forwarded for classification.

22. (Currently Amended) A system for classifying packets based on packet content, the system comprising:

a lexical analyzer operable to determine the characters of a packet;

a content engine interfaced with the lexical analyzer and operable to determine if the packet characters match a predetermined expression;

a hash engine interfaced with the lexical analyzer and the content engine, the hash engine operable to determine a hash for a subexpression of an expression match;

a hash look-up table interfaced with the hash engine to determine if the hash matches a predetermined string; and

a tag map operable to perform mappings between expressions and subexpressions to generate modified tags for packets according to the predetermined expressions found by the content engine ~~that classifies the packet with a tag according to expression and subexpression matches~~.

23. (Original) The system of Claim 22 wherein the hash look-up table comprises a high bit engine that indexes the hash by high bits and a low bit engine that matches low bits to strings determined by the high bit index.

24. (Original) The system of Claim 22 wherein the content engine comprises non-deterministic finite automata encoded with field programmable gate arrays.

25. (Original) The system of Claim 22 further comprising a state store module interfaced with the content engine, the state store module storing the state of the content engine for a first packet of a first stream and loading the content engine with the stored state when the next packet of the first stream is searched by the content engine.

26. (Original) The system of Claim 25 further comprising a stream retriever that sends the first stream identification to request the next packet of the first stream and determines the stream identification of the received next packet to determine if the next packet is associated with the first stream.

27. (Original) The system of Claim 26 wherein the stream retriever instructs the state store module to store the first stream packet's state if the next packet received is associated with a second stream.

28. (Original) The system of Claim 22 wherein the content engine determines an expression match by completing plural states, with one state associated with each character of the expression.